	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD	
--	--	--

EXE

	RRRRRRRR RR		
	\$		

```
%TITLE 'EDTSLWRITE - write to a file'
MODULE EDTSLWRITE (
                IDENT = 'V04-000'
```

! Write to a file ! File: LWRITE.BLI Edit: JBS1058

BEGIN

COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. ALL RIGHTS RESERVED.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY TRANSFERRED.

THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.

FACILITY: EDT -- The DEC Standard Editor

ABSTRACT:

This module executes the line mode EXIT and WRITE commands.

ENVIRONMENT: Runs at any access mode - AST reentrant

AUTHOR: Bob Kushlis, CREATION DATE: February 3, 1978

MODIFIED BY:

1-001 - Original. DJS 30-JAN-1981. This module was created by extracting the routines WRITE_FILE, EDT\$\$EXI_CMD, and EDT\$\$WR_CMD from the module EXEC.BLI.

1-002 - Regularize headers. JBS 23-Mar-1981

1-003 - Change WRITE_COM to EDT\$\$WR_CMD. JBS 30-Mar-1981

1-004 - Use the new message codes. JBS 04-Aug-1981

1-005 - Don't write out summary when flag clear. STS 05-0ct-1981

1-006 - Convert open output file to use EDT\$FILEIO. STS 13-Jan-1982

1-007 - Remove division from line number calculations. SMB 15-Jan-1982

1-008 - Convert reads and writes to use EDT\$FILEIO. STS 15-Jan-1982

1-009 - Add capability for 15 digit line numbers. SMB 19-Jan-1982

1-010 - pass sequence numbers by descriptors. STS 20-Jan-1981

1-011 - Change the checking of sequence/nosequence. STS 21-Jan-1982

1-012 - Fix size bugs related to 15 digit sequences & increments. SMB 26-Jan-1982

```
EDTSLWRITE
                                                                                                                                                                                       | 1-013 - Remove original Line numbers. SMB 28-Jan-1982 |
|-014 - Fix bug in writing with /SEQ. SMB 30-Jan-1982 |
|-015 - Add range checking on seq.# and incr. SMB 5-feb-1982 |
|-016 - Take out call to edt$Sqet fnam. STS 10-feb-1982 |
|-017 - Change line number of vivision to a routine call. SMB 11-feb-1982 |
|-018 - Format filename for output. STS 12-feb-1982 |
|-019 - Pass filename for output. STS 12-feb-1982 |
|-020 - Add Literals for callable EDT. STS 08-Mar-1982 |
|-020 - Add Literals for callable EDT. STS 08-Mar-1982 |
|-021 - Use the input file name. JBS 23-Mar-1982 |
|-022 - Take out call to edt$ 1-022 |
|-024 - Minor edits, plus remove code that doesn't seem to be executed. SMB 30-Mar-1982 |
|-025 - Correct a type in name is stored so we get "real" name instead of "temp" filename. SMB 31-Mar-1982 |
|-026 - Add code for CONTROL C. check during record writing. SMB 08-Apr-1982 |
|-027 - Print messages on CLOSE errors. JBS 12-Apr-1983 |
|-028 - Convert pDP-11 command file names to uppercase. SMB 13-Apr-1982 |
|-029 - Clear PREV RANGE when creating a range block for EXIT. JBS 15-Apr-1982 |
|-030 - Don't set up any message for nosummary. STS 16-Apr-1982 |
|-031 - Change NO FILE error message and processing for WRITE. SMB 21-Apr-1982 |
|-032 - Save original buffer address. STS 10-May-1982 |
|-033 - Make minor modifications based on code review input. SMB 24-May-1982 |
|-035 - Fix bug with explicit file names. SMB 25-May-1982 |
|-035 - Fix bug with explicit file names. SMB 25-May-1982 |
|-036 - Stop processings and selecting based on code review input. SMB 24-May-1982 |
|-037 - Make minor modifications based on code review input. SMB 24-May-1982 |
|-038 - Stop processing and processing for WRITE. SMB 21-Apr-1982 |
|-037 - Pass default file name using RMB parameter. JBS 15-Jun-1982 |
|-037 - Dass default file name using RMB parameter. JBS 15-Jun-1982 |
|-038 - Stop processings and selecting smb and processing for write errors. STS 21-Jul-1982 |
|-046 - Fix up the references to the EDTSM symbols. JBS 23-Aug-1982
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                                                                                                                      EDT$LWRITE - write to a file
                        0096
0097
0098
                                                                                                                         0099
                                                                                                                        0100
                                                                                                                        0101
                                                                                                                      0102
                                                                                                                        0104
                                                                                                                        0106
                                                                                                                                                                                                                                                           Because of search lists on VMS, don't use the input file name. JBS 29-Jul-1983
                      108
                                                                                                                        0108
```

```
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
EDTSUBRITE
                           EDT$LWRITE - write to a file
                                                                                                                                                     VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                            Declarations
                                         "SBTIL 'Declarations'
                           TABLE OF CONTENTS:
     115678901234567890123133678901244567
                                         REQUIRE 'EDTSRC: TRAROUNAM';
                                         FORWARD ROUTINE
                                               WRITE FILE,
EDT$$EXI CMD : NOVALUE,
EDT$$WR_CMD : NOVALUE;
                                                                                                                             Write an output file
Process the EXIT command
Process the WRITE command
                                            INCLUDE FILES:
                                         REQUIRE 'EDTSRC:EDTREQ';
                                         XIF XBLISS (BLISS32)
                                         THEN.
                                         REQUIRE 'EDTSRC:SYSSYM';
                                         %FI
                                         LIBRARY 'EDTSRC: SUPPORTS':
                                            MACROS:
                                                      NONE
                                            EQUATED SYMBOLS:
                                        EXTERNAL LITERAL

EDT$M_NOOUTPUT,

EDT$K_WRITE_FILE,

EDT$K_OUTPUT_FILE,

EDT$K_OPEN_OUTPUT_SEQ,

EDT$K_OPEN_OUTPUT_NOSEQ,

EDT$K_PUT,

EDT$K_CLOSE,

EDT$K_CLOSE_DEL;
     148
149
150
151
152
153
154
157
158
159
                                            OWN STORAGE:
                                                      NONE
     160
                            0760
                           0761
                                            EXTERNAL REFERENCES:
     162
                                                       In the routines
```

```
E
```

```
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
                                   EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                                                                                                                              VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
EDISLURITE
V04-000
                                   0764
0765
0766
0767
0768
0769
0770
                                                    %SBTTL 'WRITE_FILE - write on a file'
ROUTINE WRITE_FILE (
       165
                                                                                                                                                                Write on a file
          6
                                                            RANGE,
                                                                                                                                                                Range to write
       168
                                                             CHECK
                                                                                                                                                               Do consistency checking and file name defaulting
      169
                                                             ) =
      171
172
173
                                   0771
0772
0773
0774
0775
0776
0777
0778
                                                       FUNCTIONAL DESCRIPTION:
      174
175
176
177
                                                                     This routine is used by the WRITE and EXIT commands to write an output file. The /SEQ switch determines whether sequence
                                                                     numbers should be written.
      178
179
                                                        FORMAL PARAMETERS:
      180
181
                                                          RANGE
                                                                                                        the range block for the range to be written.
                                   0780
                                   0781
0782
0783
      182
183
                                                                                                        a flag indicating that the consistency check should be done (set if we are exiting, clear otherwise). Also permits defaulting
                                                          CHECK
      184
185
                                                                                                        of the file name to the output or input file name.
                                   0784
0785
      186
187
                                                        IMPLICIT INPUTS:
                                   0786
0787
                                                                    EDTSSG_OUT_NAMLEN
EDTSSA_OUT_NAM
EDTSSG_INP_NAMLEN
EDTSSA_INP_NAM
EDTSSA_INP_NAM
EDTSSA_CUR_BUF
EDTSSL_IO_VFCHD
EDTSSL_LNO
EDTSSL_LNO
EDTSSA_FMT_CUR
EDTSSA_FMT_CUR
EDTSSA_FMT_CUR
EDTSSA_FMT_CUR
EDTSSA_FMT_CUR
EDTSSA_FMT_CUR
EDTSSA_SUMRY
EDTSSA_SUMRY
EDTSSA_SUMRY
EDTSSA_EXE_CURCMD
EDTSSA_EXE_CURCMD
EDTSSG_EXE_SBITS
EDTSSG_EXE_SBLK
EDTSSG_WRITE_MSG
EDTSSG_ABT_WRITE
      188
189
                                   0788
      0789
                                   0790
0791
0792
0793
                                  0794
0795
                                  0796
0797
0798
0799
                                   0800
                                  0801
0802
0803
0804
0805
0806
0807
                                                        IMPLICIT OUTPUTS:
                                   0809
0810
0811
0812
0813
0814
0815
0816
0817
0818
                                                                     EDT$$G_EXE_SBITS
EDT$$G_CC_DONE
EDT$$G_WRITE_MSG
EDT$$G_ABT_WRITE
                                                        ROUTINE VALUE:
                                                                     0 = no write took place
1 = file written successfully
                                                        SIDE EFFECTS:
```

```
EDTSLWRITE
                              EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                                                        16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
                                                                                                                                                                    VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
     Types an error message if the file is not properly written.
                                                            If the consistency check fails, arranges to save the journal file.
                                                    BEGIN
                                                   EXTERNAL ROUTINE

EDT$$SC_SETWID,
EDT$$SC_POSCSIF,
EDT$$STOP_WKINGMSG,
EDT$$CHK_CC,
EDT$$LDIV,
EDT$$CALLFIO,
EDT$$FMT_CH,
EDT$$FMT_STRCNT : NOVALUE,
EDT$$FMT_CRLF,
EDT$$FMT_MSG,
EDT$$RNT_LNRNG,
EDT$$RNG_REPOS,
EDT$$FIOPN_ERR;
                                                                                                                                          Reset terminal width
                                                                                                                                          Absolute cursor positioning
                                                                                                                                         Stop working message check for CONTROL/C typed
                                                                                                                                       ! sets up handler and calls general file routine ! Put a character in the format buffer
                                                                                                                                       ! Terminate an output line
                                                                                                                                       ! Put a message in the format buffer
                         ! Print a file I/O error
                                             XIF XBLISS (BLISS32)
                                                    EXTERNAL ROUTINE EDT$$FMT_STR. STR$FREET_DX:
                                                                                                                                      ! deallocates dynamic descriptors
                                             XELSE
                                                    EXTERNAL ROUTINE
                                                            EDT$$GET_FNAM;
                                                   EXTERNAL EDT$$G_INP_NAMLEN, EDT$$A_INP_NAM;
                                                                                                                                      ! Length of input file name ! Address of input file name
                                             XF I
                                                   EXTERNAL

EDT$$G_TI_WID,
EDT$$G_ABT_WRITE,
EDT$$G_SAV_TIWID,
EDT$$G_MESSAGE_LINE,
EDT$$A_IO_FNAM,
EDT$$G_EXT_MOD,
EDT$$G_OUT_NAMLEN,
EDT$$A_OUT_NAM,
EDT$$G_RNG_FRSTLN,
EDT$$Z_RNG_ORIGPOS : POS_BLOCK,
EDT$$V_OPTIONS : BITVECTOR [32],
EDT$$A_CUR_BUF : REF_TBCB_BLOCK,
EDT$$L_IO_VFCMAX : LN_BLOCK,
                                                                                                                                      ! Terminal's width
                                                                                                                                      1 = abort output
                                                                                                                                       ! Length of output file name ! Address of output file name
                                                                                                                                      ! bit 3 = /NOOUTPUT
                                                                                                                                     ! Value 6.5535 * (10**9)
```

```
G 1
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
EDTSLWRITE
                              EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                                                                                                       VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                                                             EDT$$L_LNOO : LNOVECTOR [14],
EDT$$A_FMT_CUR,
EDT$$G_FMT_LNPOS,
EDT$$T_FMT_BUF,
      ! powers of ten
                                                                                                                                         ! current position in format buffer
! address of format buffer
                             %IF SUPPORT_WPS
                                              XTHEN
                                                             EDT$$G_SUMRY,
                                                                                                                                      ! flag indicating whether to type out summary
                                              XF I
                                                            EDT$$Z_EOB_LN,
EDT$$A_WK_EN : REF LIN_BLOCK,
EDT$$A_EXE_CURCMD : REF NODE_BLOCK,
EDT$$G_EXE_SBITS,
EDT$$Z_EXE_SBLK : REF NODE_BLOCK,
EDT$$G_CC_DONE,
EDT$$G_WRITE_MSG;
                                                                                                                                            Pointer to the current command. The options switches.
                                                                                                                                            The option switch value block.
Set to 1 if control C actually aborts something
                                                                                                                                            Message to print; 1 = no message
                                                     MESSAGES ((NOFILSPC, NOFILWRT, CONCHKFLD, ERROUTFIL, WRIFILCRE, OUTFILCRE, SEQNUMOV, SEQINCROV, OUTFILCL
                                                             WRIFILCLO)):
                                                     LOCAL
                                                             CLOSEMSG,
                                                                                                                                             File close message
                                                                                                                                            File open message code for what kind of open code for which kind of file descriptor for file name descriptor for header info
                                                             OPNMSG,
                                                            FILECODE,
FILESTRM,
FILE DESC: BLOCK [8, BYTE],
RHB DESC: BLOCK [8, BYTE],
STAT DESC: BLOCK [8, BYTE],
STATOS,
                                                                                                                                             descriptor for puts
                                                                                                                                            Status of open or close attempt
Number of lines written.
                                                            L_COUNT: LN_BLOCK,
C_COUNT,
NAME,
NAME_LEN,
DIGIT,
                                                                                                                                             Number of characters written
                                                                                                                                            file name string pointer 
file name string length
                                                                                                                                            Holds integer result from division 48-bit line number
                                                             LINNO : LN_BLOCK,
                                                             SEQ. NUM,
SEQ_INCR;
                                                                                                                                            Flag indicating sequenced
                                                                                                                                             Current sequence number
                                                                                                                                         ! Sequence number increment
                             0918
0919
0920
                                              XIF XBLISS (BLISS32)
                                              THEN
                                                     RHB_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
RHB_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
STAT_DESC [DSC$B_DTYPE] = DSC$R_DTYPE_T;
STAT_DESC [DSC$B_CLASS] = DSC$K_CLASS_S;
FILE_DESC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
FILE_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
                               0921
0922
0923
0924
0925
0926
0927
0928
0931
0931
0933
                                             XF I
                                                     RHB_DESC [DSC$A_POINTER] = 0;
RHB_DESC [DSC$W_LENGTH] = 0;
FILE_DESC [DSC$A_POINTER] = 0;
FILE_DESC [DSC$W_LENGTH] = 0;
                                                      NAME_LEN = 0;
```

```
EDTSLWRITE
V04-000
                    EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                               16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
                                                                                                            VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                             Position to the front of the range.
                   EDT$$G_RNG_FRSTLN = 1;
EDT$$CPY_MEM (POS_SIZE, .EDT$$A_CUR_BUF, EDT$$Z_RNG_ORIGPOS);
   IF ( NOT EDT$$RNG_REPOS (.RANGE)) THEN RETURN (0);
                                Check for an explicit file specification.
                                   IF (.EDT$$A_EXE_CURCMD [FSPCLEN] NEQ 0)
                                   THEN
                                       BEGIN
                                       NAME = .EDT$$A_EXE_CURCMD [FILSPEC];
NAME_LEN = .EDT$$A_EXE_CURCMD [FSPCLEN];
                              ! If no file name is specified and this is EXIT, use the output file name.
                                   IF ((.CHECK NEQ 0) AND (.NAME_LEN EQL 0))
                                   THEN
                                       BEGIN
                                        NAME = .EDT$$A_OUT_NAM;
                                        NAME_LEN = .EDT$$G_OUT_NAMLEN;
                                On VMS, because of search lists, it is possible for an input open to resolve
                                to a different directory than an output open with the same file name. Therefore, if we want the output file to be returned to the same directory as the input file
                                we must depend on using the resultant file name from the input open to specify where to put the output file, rather than being able to just re-use the input
                                At this point we reject the file specification only if it is empty and either
                                this is a WRITE command or /NOOUTPUT has been specified (or implied by /READ_ONLY).
                                   IF ((.NAME_LEN EQL 0) AND ((.CHECK EQL 0) OR ((.EDT$$V_OPTIONS AND EDT$M_NOOUTPUT) NEQ 0)))
                                   THEN
                                       BEGIN
EDT$$FMT_MSG (EDT$_NOFILSPC);
                                        RETURN (0);
                                        END:
                                PDP-11 systems do not have search lists, so we can use the input file name if the
                                output name is empty. Doing this simplifys the process of constructing the printable
                                form of the name for the summary message.
                              XIF ( NOT XBLISS (BLISS32))
XTHEN
                                   IF (.NAME_LEN EQL 0)
```

```
EDTSLWRITE
                                                                                                                                        VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER: [EDT.SRC]LWRITE.BLI; 1
                         EDT$LWRITE - write to a file WRITE_FILE - write on a file
                        0993
0993
0995
0996
0996
0999
1000
1002
1006
1006
1007
1008
1011
1016
1016
1017
1018
1019
    394567899012340067890102340406784090
                                            THEN
                                                  BEGIN
                                                  NAME = .EDT$$A INP_NAM;
NAME_LEN = .EDT$$G_INP_NAMLEN;
                                     %FI
                                       Check for the /SEQ option.
                                            IF (SEQ = .EDT$$G_EXE_SBITS<OPB_SEQ>)
                                            THEN
                                                 BEGIN
                                      ! Determine the actual sequence start and increment by reducing them by 10**5
                                                 SEQ_NUM = 0;
SEQ_INCR = 0;
MOVELINE (EDT$$Z_EXE_SBLK [SW_VAL1], LINNO);
    412
413
414
415
416
417
                                                                                                                        ! Starting Value
                                      ! Error checks for sequence number and increment (must be less than 65536)
                                                  IF (CMPLNO (LINNO, EDT$$L_LNO_VFCMAX) GTR 0)
                                                  THEN
    BEGIN
                                                        EDT$$FMT_MSG (EDT$_SEQNUMOV);
RETURN (0)
                         1020
                         1021
1022
1023
1024
1025
1026
1027
1028
1039
1031
1033
1034
1035
1038
                                                        END:
                                                 DECR I FROM 9 TO 5 DO
                                                        BEGIN
                                                        SEQ NUM = .SEQ NUM*10;
EDT$$LDIV (LINNO, DIGIT,
                                                        SEQ_NUM = .SEQ_NUM + .DIGIT;
                                                 MOVELINE (EDT$$Z_EXE_SBLK [SW_VAL2], LINNO);
                                                                                                                            ! Increment
                                                  IF (CMPLNO (LINNO, EDT$$L_LNO_VFCMAX) GTR 0)
                                                  THEN
                                                        BEGIN
                                                        EDT$$FMT_MSG (EDT$_SEQINCROV);
                                                        RETURN (0)
                                                        END:
                         1039
                         1040
1041
1042
1043
1044
1045
1046
1047
                                                 DECR 1 FROM 9 TO 5 DO
                                                        BEGIN
                                                        SEQ_INCR = .SEQ_INCR*10;
EDT$$LDIV (LINNO, DIGIT, .I);
SEQ_INCR = .SEQ_INCR + .DIGIT;
                                                 END:
```

ED VO

```
EDTSLWRITE
                       EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                             16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
                                                                                                                                VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
    ! Try to open the file and set up file specific messages
                                        IF (.CHECK EQL 0)
                                              BEGIN

FILESTRM = EDT$K WRITE_FILE;

CLOSEMSG = EDT$ WRIFILCLO;

OPNMSG = EDT$_WRIFILCRE;
                                              END
                       1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
                                        ELSE
                                              BEGIN
                                              FILESTRM = EDTSK OUTPUT FILE;
CLOSEMSG = EDTS OUTFILCEO;
                                              OPNMSG = EDTS_OUTFILCRE:
                                        IF (.SEQ NEQ O) THEN FILECODE = EDT$K_OPEN_OUTPUT_SEQ ELSE FILECODE = EDT$K_OPEN_OUTPUT_NOSEQ;
                                        STRING_DESC (FILE_DESC, NAME_LEN, .NAME);
STATUS = EDT$$CALCFIO (.FILECODE, .FILESTRM, FILE_DESC, RHB_DESC);
                       1072
                                         IF .STATUS
                       1074
1075
                                              BEGIN
                                              MOVELINE (EDT$$L_LNO_ZERO, L_COUNT);
                       1076
1077
1078
1079
                                               C COUNT = 0:
    478
                                              EDT$$G_ABT_WRITE = 0;
    480
481
482
483
484
486
487
488
490
491
                                              WHILE (EDT$$NXT_LNRNG (0) AND (.EDT$$A_WK_LN NEQA EDT$$Z_EOB_LN) AND ( NOT .EDT$$G_ABT_WRITE)) DO
                       1080
                       1081
1082
1083
1084
1085
1086
                                     Check for a CONTROL/C. If one has been typed abort the write operation.
                                                    IF EDT$$CHK_CC ()
                                                    THEN
                                                          BEGIN
                       1088
                                                          EDT$$G_CC_DONE = 1;
EDT$$G_ABT_WRITE = 1;
                       1089
                       1090
    492
                       1091
                                                    ELSE
                       1092
1093
                                                          BEGIN
    494
                       1094
1095
                                   ! Set up the RHB if /SEQ was used.
    496
                       1096
                                                          IF (.SEQ NEQ 0)
    498
    499
                       1098
    500
501
                        1099
                                                                IF (.EDT$$Z_EXE_SBLK [SEQ_VAL] EQL 0)
THEN
                       1100
    502
503
504
505
506
                       1101
                       1102
                                     If no sequence start was given, then use the buffer's current line numbers in the VFC field (/SEQ)
```

ED'

:

```
EDT$LWRITE
                        EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                                                                     VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER: [EDT. SRC]LWRITE.BLI; 1
                                                                         MOVELINE (EDT$$A_WK_LN [LIN_NUM], LINNO);
    DECR I FROM 14 TO 10 DO
                                                                               WHILE (CMPLNO (LINNO, EDT$$L_LNOO [.1]) GEQ 0) DO SUBLINE (EDT$$L_LNOO [.1], LINNO);
                                                                         IF (CMPLNO (LINNO, EDT$$L_LNO_VFCMAX) GEQ 0)
                                                                               EDT$$L_IO_VFCHD = 65535
                                                                         ELSE
                                                                               BEGIN
                                       Set up sequence numbers, but first divide by 10**5
                                                                               EDT$$L_IO_VFCHD = 0;
                                                                               DECR I FROM 9 TO 5 DO
                                                                                     BEGIN
                                                                                     EDT$$L_IO_VFCHD = .EDT$$L_IO_VFCHD*10;
EDT$$LDIV (LINNO, DIGIT, .I);
EDT$$L_IO_VFCHD = .EDT$$L_IO_VFCHD + .DIGIT;
                                                                               END
                                                                         END
                                                                   ELSE
                                      Otherwise, use the given sequence start and increment (/SEQ:st:inc)
                                                                        BEGIN
EDT$$L IO_VFCHD = .SEQ_NUM;
SEQ_NUM = .SEQ_NUM + .SEQ_INCR;
                                                                         IF (.SEQ_NUM EQL 65535) THEN SEQ_INCR = 0;
                                                                         END:
                                      Write a line to the file.
                                                            RHB_DESC [DSC$W_LENGTH] = 2;

RHB_DESC [DSC$A_POINTER] = EDT$$L_IO_VFCHD;

STAT_DESC [DSC$D_LENGTH] = .EDT$$A_WR_LN [LIN_LENGTH];

STAT_DESC [DSC$A_POINTER] = EDT$$A_WK_LN [LIN_TEXT];

STATOS = EDT$$CAELFIO (EDT$K_PUT, .FIEESTRM, STAT_DESC, RHB_DESC);
                                                             IF ( NOT .STATUS)
                                                             THEN
                                                                   BEGIN
                        1158
1159
1160
                                    XIF XBLISS (BLISS32)
                        1161
                                                                   EDT$$FIOPN_ERR (EDT$_ERROUTFIL, .EDT$$A_IO_FNAM);
                                    XF I
```

VO.

```
VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER: [EDT. SRC] LWRITE.BLI; 1
EDT$LWRITE
                   EDT$LWRITE - write to a file WRITE_FILE - write on a file
                    1163
1164
1165
1166
1167
1168
1169
1170
   EDT$$G_ABT_WRITE = 1;
                                                 ELSE
                                                      BEGIN
                               Count lines and characters for consistency check.
                                                     ADDLINE (NUMBER ONE, L_COUNT);
C_COUNT = .C_COUNT + .EDT$$A_WK_LN [LIN_LENGTH];
                                                 END;
                                            END:
                               Watch for the write abort flag, which can be set if we have an error on the input file,
                               by control C or by an error writing the output file.
                                       IF .EDT$$G_ABT_WRITE
                                       THEN
                                            STATUS = EDT$$CALLFIO (EDT$K_CLOSE_DEL, .FILESTRM, FILE_DESC, 0);
                                            IF ( NOT .STATUS) THEN EDT$$FIOPN_ERR (.CLOSEMSG, FILE_DESC);
                                            EDT$$FMT_MSG (EDT$_NOFILWRT);
                   1193
                             XIF XBLISS (BLISS32)
                   1194
                             %THEN
                   1195
                                            STR$FREE1_DX (FILE_DESC);
                             %FI
                    1197
                                            RETURN (0);
                                            END:
                               Do the consistency check.
                                       IF (.CHECK NEQ 0) THEN
                                            IF ( NOT (LINNOEQL (L_COUNT, EDT$$A_CUR_BUF [TBCB_LINE_COUNT])) OR ! (.C_COUNT NEQ .EDT$$A_CUR_BUF [TBCB_CHAR_COUNT]))
                                            THEN
                                                BEGIN
EDT$$FMT_MSG (EDT$_CONCHKFLD);
                                                 EDT$$G_EXE_SBITS = (.EDT$$G_EXE_SBITS OR OPT_SAVE);
                               Close the output file and print a message giving number of lines written
                                to the file.
```

ED'

: 1

```
EDT$LWRITE
                   EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                                                                                        VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER: [EDT. SRC]LWRITE.BLI; 1
   IF (.EDT$$G_EXT_MOD) THEN EDT$$STOP_WKINGMSG ();
                                      STATUS = EDT$$CALLFIO (EDT$K_CLOSE, .FILESTRM, FILE_DESC, 0);
                                      IF .STATUS
                                      THEN
                                          BEGIN
                              Reset the screen width on EXIT if necessary. If the screen was reset, then
                              reposition the cursor at the bottom of the screen.
                                          IF (.CHECK NEQ 0)
                                               IF (.EDT$$G_TI_WID NEQ .EDT$$G_SAV_TIWID)
                                                   BEGIN
EDT$$SC_SETWID (.EDT$$G_SAV_TIWID);
EDT$$SC_POSCSIF (.EDT$$G_MESSAGE_LINE, 0);
                              Print a message if one is requested. This will be the 'input file does not have standard format'
                              message deferred because we thought the user was never going to write the buffer.
                                          IF (.EDT$$G_WRITE_MSG NEQ 1)
                                          THEN
                                               BEGIN
EDT$$FMT_MSG (.EDT$$G_WRITE_MSG);
                                               EDT$$G_WRITE_MSG = 1;
                            XIF SUPPORT_WPS
                                          IF .EDT$$G_SUMRY
                            XF I
                                               BEGIN
                              Extract the resultant filename of the opened file and format it to write out if the summary flag is set.
                            XIF XBLISS (BLISS32)
                            XTHEN
                                               EDT$$FMT_STR (.FILE_DESC [DSC$A_POINTER], .FILE_DESC [DSC$W_LENGTH]);
                            XELSE
                                               EDT$$GET_FNAM ();
                            %FI
                                               EDT$$FMT_CH (' ');
EDT$$FMT_STRCNT (L_COUNT, UPLIT (' line'), 5);
```

ED VO

...........

```
VAX-11 Bliss-32 V4.0-742 Page DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
EDT$LWRITE
                  EDT$LWRITE - write to a file WRITE_FILE - write on a file
                                             EDT$$FMT_CRLF ()
END;
   On VMS deallocate all dynamic descriptors used
                           XIF XBLISS (BLISS32)
                                        STR$FREE1_DX (FILE_DESC);
                           %FI
                                        RETURN (1);
                                        END
                                    ELSE
                                        BEGIN
                             File was not closed, output error message.
                                        EDT$$FIOPN_ERR (.CLOSEMSG, FILE_DESC);
                             On VMS deallocate all dynamic descriptors used
                          XIF XBLISS (BLISS32)
                                        STR$+REE1_DX (FILE_DESC);
                                        RETURN (0);
                                        END
                                    END
                               ELSE
                                   BEGIN
                           ! File was not created, output error message.
                                   EDT$$FIOPN_ERR (.OPNMSG, FILE_DESC);
                             On VMS deallocate all dynamic descriptors used
                          XIF XBLISS (BLISS32)
                                    STR$FREE1_DX (FILE_DESC);
                           XF I
                                    RETURN (0);
                                    END
                                                                                ! of routine WRITE_FILE
                               END:
                                                                                           EDT$LWRITE EDT$LWRITE - write to a file \v04-000\
```

**

MOVL

MOVC3

0000000G

00000000G 00

EDT\$LWRITE V04-000	EDT\$LWRI WRITE_FI	TE	- write to - write on	a f	ile			16	-Sep-	1984 01:03 1984 12:23	:36	VAX-11 Bliss-32 V4.0-742 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1	age 15
			0000000G	00	04	AC 01 50	DD FB	0003A 0003D 00044 00047		PUSHL CALLS BLRS	RANGE #1. E RO. 1 68\$	DT\$\$RNG_REPOS	: 0940
					000000000	484 00 A0 09	31 DO DS	0004A 00051	1\$:	BRW MOVL TSTL	68\$ EDT\$\$	A_EXE_CURCMD, RO	0946
			08	SA AE 56	08 00 08 04	09 A0 AC AE 56	00 00 00 04	0005A 0005F 00063	2\$:	PUSHL CALLS BLBS BRW MOVL TSTL BEQL MOVL MOVL CLRL TSTL	8(R0) 12(R0	NAME NAME_LEN	0949 0950 0957
					04 08	17 AE AE OF	13 06 05	00066 00068 0006A 0006D 00070		BEQL INCL TSTL BNEQ	4(SP) NAME	LEN	
			08	5A AE	00000000G 0000000G 08	00 00 AE 19 56	DO D	00072 00079 00081 00084 00086	3\$:	BEQL INCL TSTL BNEQ MOVL MOVL TSTL BNEQ TSTL	EDTS! EDTS! NAME. 5\$ R6	BA_OUT_NAM, NAME BG_OUT_NAMLEN, NAME_LEN _LEN	0960 0961 0975
			0000000G	8F	000000006	00 00 80	13 03 13	00088 0008A 00095		BEQL BITL BEQL PUSHL BRB EXTZV	45	SV_OPTIONS, #EDTSM_NOOUTPUT	
					0000000G	8F 52	DD 11		45:	PUSHL	WEDTS	_NOFILSPC	0978
	57 00000000G	00		01		57	EF E8	0009F 000A8	5\$:	RIBS	SEO.	#1, EDT\$\$G_EXE_SBITS, SEQ	: 1004
					0	OCE 59	04	000AB	6\$:	BLBS BRW CLRL CLRL MOVL MOVC3	SEQ_N	NUM	1010
	10	AF	00	50	000000006	5B	00	000B0 000B2 000B9		MOVL	EDTS:	Z EXE SBLK, RO	1010 1011 1012
	10	AE	08	50	000000006	00 AE OF	3C B1	000BF		MUVZWL	HIGH HIGH	2, RÓ 1, RO	1017
	,			50 50	000000006	18 00 AE 05	12	000C6 000CC 000CE 000D5 000D9 000DB 000DE 000E2 000E4 000E9 000EB		CMPW BLSSU BNEQ MOVL CMPL BGEQU MNEGL BRB BNEQ CLRL BRB MOVL BLEQ PUSHL BRB MOVL PUSHAB PUSHAB CALLS ADDL2 ACBB	9\$ LOW_2	NUM INCR BZ_EXE_SBLK, RO B_CRO I, RO I, RO RO RO RO S_SEQNUMOV I SEQ_NUM	
				50		05	1E CE	000D9 000DB	7\$:	BGEQU	8\$ #1. F	80	
						09	11	000DE 000E0	8\$:	BRB BNEQ	10\$		
				50		03	11	000E2	oe.	CLRL BRB	10\$		
				50	000000006	08 8F	15	000E9	9\$: 10\$:	BLEQ	12\$	S SEQNUMOV	1020
				52	00000000	5F 09	DD 11 DO	000F1 000F3	11\$: 12\$: 135:	BRB MOVL	18\$	_ seamonov	:
				52		0A 52	C4 DD 9F	000F6 000F9	135:	MULL2 PUSHL	#10. I	SEQ_NUM	1024 1026 1027
			00000000	00	10 18	AE	9F	000FB		PUSHAB	LINN) COTES! OTV	
FF	F3	52	0000000G	00 59 8F	ОС	OA 52 AE 03 AE 05	FB CO 9D	000F3 000F6 000F9 000FB 000FE 00101 00108		ADDL2	DIGI	TO DT\$\$LDIV T, SEQ_NUM V-1, I, 13\$	1028 1024

DT\$LWRITE	EDTSLWR	ITE :	- write to	a f	ile			16-Sep 14-Sep	-1984 01:03 -1984 12:23	3:36 VAX-11 Bliss-32 V4.0-742 B:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1	Page 16
	10	AE	14	50 50 50	00000000G 00000000G 14	00 06 00 AE 0F	DO 001 28 001 3C 001 B1 001 1F 001 12 001 D0 001	13 1A 20 27	MOVL MOVZWL CMPW BLSSU BNEQ MOVL CMPL BGFQU MNEGL	EDT\$\$Z_EXE_SBLK, RO #6, 20TRO), LINNO HIGH_2, RO HIGH_1, RO 14\$	1031
				50		18 00 AE 05	1E 001	3A	BNEQ MOVL CMPL BGFQU	16\$ LOW_2, RO LOW_1, RO 15\$ #1, RO 17\$	
				50		01 09 04 50	CE 001 11 001 12 001 04 001	3C 14\$:	BNEA	#1 R0 17\$ 16\$ R0 17\$	
				50	0000000G	01	DO 001	47 16\$: 4A 17\$:	BRB MOVL BLEQ PUSHL CALLS	#1 R0	1036
			0000000G	00		8F 01 372	DD 001 FB 001 31 001	59	DDII	#1 EDT\$\$FMT_MSG	•
				52 5B		09	DO 001	5C 19\$:	MOVL MULL2	WEDTS_SEQINCROV W1, EDTSSFMT_MSG 685 W9, I W10, SEQ_INCR	1037 1040 1042 1043
					10 18	AE	DD 001 9F 001 9F 001	62 64 67	PUSHAB	DIGIT	1043
			0000000G	00 5B 8F	00	0A 52 AE 03 AE 05	FB 001	6A 71	MOVL MULL2 PUSHAB PUSHAB CALLS ADDL2 ACBB TSTL BNEQ	#3, EDT\$\$LDIV DIGIT, SEQ_INCR	1044
FFE3		52	FF	8F		05 56 17	9D 001	75 70 21 \$:	TSTL	#5, #-1, I, 20\$ R6	1044 1040 105
				56 58 53	00000000G 00000000G 00000000G	8F 8F 8F 15	12 001 00 001 00 001 11 001	80 87	MOVL MOVL BRB	#3, EDT\$\$LDIV DIGIT, SEQ_INCR #5, #-1, I, 20\$ R6 22\$ #EDT\$K WRITE_FILE, FILESTRM #EDT\$_@RIFILCLO, CLOSEMSG #EDT\$_WRIFILCRE, OPNMSG 23\$	1056 1057 1058
						8F	DO 001	8E 95	MOVL BRB	WEDTS_WRIFILCRE, OPNMSG	: 1058
				56 58 53	00000000G 00000000G 00000000G	8F 8F	DO 001 DO 001 DO 001	97 22\$: 9E 45	MOVL	WEDTS OUTFILCEO, CLOSEMSG	106 106 106 106
				,,	***************************************	6E 57	D4 001 D5 001 13 001	AC 235:	MOVL CLRL TSTL BEQL INCL	WEDTS_OUTFILCRE, OPNMSG (SP) SEQ 24\$	1067
				52	00000000	OB 6E	D6 001	B2	BEQL	(SP)	
					00000000G	07 8F	DO 001 11 001 DO 001	BB BD 24\$:	MOVL BRB MOVL	#EDT\$K_OPEN_OUTPUT_SEQ, FILECODE #EDT\$K_OPEN_OUTPUT_NOSEQ, FILECODE	
					0C 38	5A AE	DD 001 9F 001 9F 001	(4 258:	PUSHL	NAME LEN	1069
			0000000G	00		O3	DO 001 DD 001 9F 001 9F 001 9F 001 9F 001	C9 CC	CALLS	#3. STR\$COPY_R	1070
					28 34 0044	AE 8F	9F 001	D9	PUSHAB	FILE DESC MAM <r2,r6></r2,r6>	
			0000000G	57		AE 03 AE 8 04 0 5 7	FB 001	DD F4	MOVL PUSHAB PUSHAB CALLS PUSHAB PUSHAB PUSHAR CALLS MOVL BLBS BRW	NAME NAME_LEN FILE_DESC #3, STR\$COPY_R RHB_DESC FILE_DESC #^M <r2,r6> #4, EDT\$\$CALLFIO R0, STATUS STATUS, 26\$</r2,r6>	1073
	18	AF	0000000G	03	0	2CB	E8 001 31 001 28 001 04 001	EA ED 26\$:	BRW MOVC3		1072
		7.	20000000	30	000000006	06 5A 00	D4 001	F6	CLRL	#6, EDT\$\$L_LNO_ZERO, L_COUNT C_COUNT EDT\$\$G_ABT_WRITE	1075 1076 1077

E.7

T\$LWRITE 4-000	EDT\$LWR WRITE_F				75	04		5-Sep-1 4-Sep-1 27\$:		-(SP)	.I;1 Page 1 ; 107
			00000000G	00	01 50	0 F E 3 9 D 1 E 3 F E D 3 E 3 D 9 1 3 D 2 D C 9 B	00200		CLRL CALLS BLBS BRW	-(SP) #1, EDT\$\$NXT_LNRNG R0, 29\$ 53\$	100
				50	00000000G 00	31 9E	0020A	28\$: 29\$:	BRW	53\$ EDT\$\$Z EOB LN. RO	
				50	00000000G 00	D1	00214 0021B		CMPL BEQL	EDT\$\$Z_EOB_LN, RO EDT\$\$A_WK_EN, RO 28\$ EDT\$\$G_ABT_WRITE, 30\$	
				03	00000000G 00	E9	0021D 00224		BLBC	EDT\$\$G_ABT_WRITE, 30\$	
			0000000G	00 0A	00 50	FB E9	00227 0022E	30\$:	CALLS	#O, EDT\$\$CHK_CC	108
			0000000G	00	01 012E	D0	00231		MOVL	#0, EDT\$\$CHK_CC R0, 31\$ #1, EDT\$\$G_CC_DONE 49\$; 108 ; 108 ; 109
				03	6E 00DE	E8	0023B 0023E	31\$:	BLBS	(SP), 32\$	109
				50	00000000G 00 00000000G 00 00000000G 00 016A 012E 00000000G 00 01 00 0000000G 00 00 00	95	00241	32\$:	MOVAB CMPL BLBC BLBC BLBC BLBC BRWLS BNOVB BRWLS BRWLS BRWLS BROVL BRWLS	(SP), 32\$ 48\$ EDT\$\$Z_EXE_SBLK, RO 1(RO)	110
					03 00BA	13	0024B 0024D		BEQL	33\$ 47\$	
	10	AE	01	50 A0	00000000G 00	28	00250	33\$:	MOVL MOVC3	EDTSSA_WK_LN, RO	110
		51		50	0Ē	00	0025D 00260	34\$:	MOVL MULL3	#14, I #6, I, R1	110
			04	51 A1	00000000G0041	9E B1	00264 00260	35\$:	MOVAB	EDT\$\$L_LNOO[R1], R1 HIGH 1, 4(R1)	
					14 AE 08 11 10 AE 05 01 09 04 52	1F	00271		BLSSU	36\$ 38\$	
				61	10 AE	D1 1E	00275		BGEQU	LOW_1, (R1) 37\$	
				52	01 09	11	0027B 0027E	36\$:	MNEGL BRB	#1, R2 39\$	
					04 52	11 12 04	00280	37\$:	BRB BNEQ CLRL	38\$ R2	
				52	01	11 00 19	00284	38\$: 39\$:	BRB MOVL	39\$ #1, R2	
					16 AE	19 B0	00289 0028B	39\$:	BLSS	40\$ UPPER_WORD, SAVE	111
			10 14 16	SE AE AE	04 A1	09	0028F 00293		SUBL2 SBWC	(R1), LINNO 4(R1), LINNO	
					52 CE	B0	00298 00290		MOVW BRB	SAVE, UPPER_WORD	111
FFBB		50	FF	8F 50 50	000000000	B0 C2 D9 B0 11 9D 3C B1	0029E	40\$:	MOVZWL	#10, #-1, I, 34\$ HIGH_2, RO	111
				50	14 AE	B1	002AC 002B0		BLSSU	HIGH_1, RO	
				50	16 AE 04 A1 52 CE 0A 00000000G 00 14 AE 0F 18 00000000G 00 10 AE 05 01 09 01	D0	001F07AD4BD47E138BE18BBE18BBE18BBE18BBE18BBE18BBE18BB		BRB MOVL BLSS MOVW SUBL2 SBWC MOVW BRB ACBB MOVZWL CMPW BNEQ MOVL CMPL BNEQ CMPL BNEQ CMPL BNEQ CLRL BRB BNEQ CLRL BRB BNEQ CLRL BRB BNEQ CLRL BRB BNEQ CLRL BNEGW	33\$ 47\$ EDT\$\$A WK_LN, RO #6, 1(RO), LINNO #14, I #6, I, R1 EDT\$\$L_NOO[R1], R1 HIGH_1, 4(R1) 36\$ LOW_1, (R1) 37\$ #1, R2 39\$ #1, R2 40\$ UPPER_WORD, SAVE (R1), LINNO 4(R1), LINNO SAVE, UPPER_WORD 35\$ #10, #-1, I, 34\$ HIGH_2, RO HIGH_1, RO 41\$ 43\$ LOW_2, RO LOW_1, RO 42\$ #1, RO 44\$ #1, RO 44\$ #1, RO 45\$ #1, EDT\$\$L_!O_VFCHD	:
					10 AE	1E	002BB 002BF		BGEQU	LOW_1, RO	
				50	01	CE	00201	41\$:	BRB	44\$	
					50	12 04 11 00 19 AE	90508	42\$:	CLRL	RO RO	
				50	03 01	00	005CV	435: 445:	MOVL	#1, RO	
			0000000G	00	09	AE	00201	445:	WNECH	#1, EDT\$\$L_IO_VFCHD	111

ED

EDTSLWRITE V04-000	EDT\$LWRITE - write to WRITE_FILE - write on	a file a file				984 01:03 1984 12:23	:36 VAX-11 Bliss-32 V4.0-742 Page :47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1	18
	0000000G	00000000G 52 00	45 00 09 0A 52	11 00208 B4 00200 D0 002E0 A4 002E0 9F 002E0 9F 002E0 9F 002E0 9F 00300 11 00300 B0 00300 C0 0031	45\$: 46\$:	BRB CLRW MOVL MULW2 PUSHL		1122 1124 1126 1127
FFDB	00000000G 00000000G 52 FF	10 18 00 00 00 8F	0A 2E A 3E 0 A 5 1 5 0	9F 002EF FB 002F A0 002F 9D 00301		BRB CLRW MOVLW2 PUSHAB PUSHAB CADDW2 ACBB MOVWAB MOVAB MOVAB MOVAB MOVAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB PUSHAB	DIGIT LINNO #3, EDT\$\$LDIV DIGIT, EDT\$\$L_IO_VFCHD #5, #-1, I, 46\$	1128 1124 1114 1139 1140
	0000000G 0000FFFF	00 59 8F	58 59 02 58	BO 0030/ CO 00314 D1 00314 12 00316 D4 00316	4/\$:	ADDL2 CMPL BNEQ CLRL	SEQ_NUM, EDT\$\$L TO_VFCHD SEQ_INCR, SEQ_NUM SEQ_NUM, #65535 48\$ SEQ_INCR	
	28 20 20 24	AE 00000000G 50 00000000G AE AE 07	59B92B200000 AAEA66F	12 00316 04 00316 96 00326 98 00336 96 00336 97 00336 97 00336 98 00336	48\$:	MOVW MOVAB MOVL MOVZBW MOVAR	(RO), STAT DESC	1149 1150 1151
	000000006	28 24 000000006	AE AE 56 8F	9F 0033E 9F 0033E DD 00341 DD 00343 FB 00349	,	PUSHAB PUSHAB PUSHL PUSHL	RHB_DESC 1 STAT_DESC 1 FILESTRM #EDT\$K_PUT	1152
		00 57 10 000000006 000000006	50 57 00 8F 02	DO 00350 E8 00353 DD 00350 FB 00363 DO 00369		MOVL BLBS PUSHL PUSHL	RO, STATUS STATUS, 50\$ EDT\$\$A_IO_FNAM #EDT\$_ERROUTFIL	1155
	0000000G 0000000G	18	01 15 AE 03	11 00370 06 00370 12 0037	50\$:	BRB INCL BNEQ		1164 1155 1171
		50 00000000G 51 5A	51	B6 00377 D0 00377 9A 00381	51\$:	INCW MOVL MOVZBL ADDL2	(RO), R1 R1, C COUNT	1172
		36 00000000g 34	7E 7E AE 56 8F	E9 0038/ 04 00391 9F 00393 DD 00398	52\$: 53\$: 54\$:	BLBC CLRL PUSHAB PUSHL	ED1330 ABI WKITE, 203	1079 1184 1187
	000000006	00000000cs 57 00	8F 04 50 57 AE	DO 003A5 E8 003A6 9F 003A6		CALLS MOVL BLBS PUSHAB	#EDTSK_CLOSE_DEL #4, EDT\$\$CALCFIO RO, STATUS STATUS, 55\$ FILE_DESC 1	189
	00000000G	00 000000006	58 02 8F 01	FB 003B0	55\$:	INCW MOVL MOVZBL ADDL2 BRW BLBC CLRL PUSHAB PUSHL CALLS MOVL BLBS PUSHAB PUSHL CALLS PUSHL CALLS BRW BLBC MOVL	#1. EDT\$\$FMT MSG	1191
	18	30 50 000000006 80	OOF D AE OO AE	DD 003B0 FB 003B0 31 003C0 E9 003C0 D0 003C0 D1 003D0	56\$:	BLBC MOVL CMPL	4(SP), 58\$ EDT\$\$A_CUR_BUF, RO 1 LOW_1, 24(RO)	195 205 208

G 2 16-Sep-1984 01:03:36 VAX-11 Bliss-32 V4.0-742 Page 19 14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 (3)

10	AO	10	OD AE 06	12 81	003D7 003D9		BNEQ	57\$ HIGH_1, 28(RO)	
1E	AO		06 5A	12	003DE 003E0		BNEQ CMPL	C_COUNT, 30(RO)	1209
		0000000G	15 8F	13	003E4 003E6	57\$:	PUSHL	#EDTS_CONCHKFLD	1212
00000000G	00	40	01 8F	DD FB 88 E9	003EC 003F3		PUSHL CALLS BISB2	#1, EDT\$\$FMT MSG #64, EDT\$\$G EXE_SBITS EDT\$\$G EXT MOD, 59\$ #0, EDT\$\$STOP_WKINGMSG	1213
0000000G	07	0000000G	8F 00 7E AE 56	E9	003FB 00402		CALLS	EDT\$\$G_EXT_MOD, 59\$ #0, EDT\$\$STOP_WKINGMSG	
		34	7E AE	94 9F	00409 0040B	598:	CLRL	#O, EDT\$\$STOP_WKINGMSG -(SP) FILE_DESC	1223
		00000000	56 8F	DD	0040E 00410		PUSHL	FILESTRM MEDISK CLOSE	
0000000G	00 57 03		04	FB DO	00416 0041D		CALLS	#4, EDT\$\$CALLFIO RO, STATUS	
	03		57 08B	E8 31 E9	00420 00423 00426		BLBS BRW	STATUS, 60\$	1225
	28 50	04	AE 00	E9 DO	00426 0042A	60\$:	BLBC	4(SP), 61\$	1233 1236
	50	00000000G	00	D1	00431		BLBC MOVL CMPL BEQL	4(SP), 61\$ EDT\$\$G_SAV_TIWID, RO EDT\$\$G_TI_WID, RO	1230
			50	13	00438 0043A		PUSHL	RO	1239
0000000G	00		01 7E 00	FB D4	0043C 00443		CALLS	#1, EDT\$\$SC_SETWID -(SP)	1240
0000000G	00	0000000G	00 02 00	DD FB	00445 0044B		PUSHL	EDTSSG_MESSAGE_LINE #2, EDTSSSC_POSCSIF	
	50	0000000G	50	D0	00452	61\$:	MOVL	EDISSG_WKITE_MSG, KU	1248
			10	13 DD	0045C 0045E		PUSHL	RO, #1 62\$ RO	1251
00000000G	00		01	FB	00460		MOVL	#1 ENTEREMT MCG	
***************************************	ŽĚ 7E	000000000	00	E9		62\$:	BLBC	#1, EDT\$\$G WRITE MSG EDT\$\$G SUMRY, 63\$ FILE DESC, -(SP) FILE DESC+4 #2, EDT\$\$FMT_STR	1252 1258 1270
000000006	00	30 38	AE O2	E9 3C DD FB	00479 00470		PUSHL	FILE DESC+4	1210
			20	DD	00483		PUSHL	#2 EDT\$\$FMT_STR	1275
0000000G	00		201 05 CF A03 0AE 01	DD FB DD 9F	00483 00485 0048C 0048E		PUSHL	#1, EDT\$\$FMT_CH	1276
		FB66 20	AE	9F	0048E 00492 00495		PUSHAB PUSHAB	P.AAA L_COUNT #3, EDT\$\$FMT_STRCNT	
00000000G	00		03	FB FB 9F	00495 0049C 004A3		CALLS CALLS PUSHAB	#3, EDT\$\$FMT_STRCNT #0, EDT\$\$FMT_CRLF	1277 1286
000000006	00	30	AE 01	9F	004A3	63\$:	PUSHAB	#O, EDT\$\$FMT_CRLF FILE_DESC #1, STR\$FREE1_DX #1, RO	
	50		01	FB 00 04 9F	004AD 004B0		MOVL RET	#1, R0	1292
		30	AE	9F	004B1	64\$:	PUSHAB	FILE DESC	1296
		30	Ó5	DD 11	004B4 004B6 004B8	458.	BRB	CLOSEMSG 66\$	1315
00000000	00	30	AE 505 AE 502 AE	9F DD FB	004RR		PUSHAB	FILE DESC OPNMSG	1313
000000006	00	30	AE	9F	004BD 004C4	67\$:	PUSHL CALLS PUSHAB	#2, EDT\$\$FIOPN_ERR FILE_DESC #1, STR\$FREE1_DX	1322
0000000G	00		01 50	FB D4	004C7 004CE	68\$:	CALLS	RO STRSFREET_DX	1328

ED VO

EDTSLWRITE

EDT\$LWRITE - write to a file WRITE_FILE - write on a file

H 2 16-Sep-1984 01:03:36 VAX-11 BLiss-32 V4.0-742 14-Sep-1984 12:23:47 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 Page (3)

04 00400

RET

; Routine Size: 1233 bytes, Routine Base: _EDT\$CODE + 0008

```
ED VO
```

```
EDT$LWRITE
                      EDT$LWRITE - write to a file
EDT$$EXI_CMD - EXIT line-mode command
                                                                                                                        VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                                 %SBTTL 'EDT$$EXI_CMD - EXIT line-mode command'
    GLOBAL ROUTINE EDT$$EXI_CMD
: NOVALUE =
                                                                                                  ! EXIT line-mode command
                                   FUNCTIONAL DESCRIPTION:
                                           Command processing routine for exit. Switch to the main buffer, write the output file and set the EDT$$G_EXITD flag if it succeeded.
                                    FORMAL PARAMETERS:
                                            NONE
                                    IMPLICIT INPUTS:
                                            EDT$$G_RCOV_MOD
                                    IMPLICIT OUTPUTS:
                                            EDT$$G_EXITD
                                    ROUTINE VALUE:
                                            NONE
                                    SIDE EFFECTS:
                                            NONE
                      1360
1361
1362
1363
1364
1365
1367
1368
1369
                                      BEGIN
                                      EXTERNAL ROUTINE
                                            EDT$$FND_BUF;
                                      EXTERNAL EDT$$G_EXITD, EDT$$G_RCOV_MOD;
                                      RANGE : NODE_BLOCK;
                                   Ignore the EXIT command encountered during recovery.
                                      IF .EDT$$G_RCOV_MOD THEN RETURN;
                                    Setup the range as WHOLE.
                                      RANGE [NODE TYPE] = RANGE NODE;
RANGE [RAN_TYPE] = RAN_WHOLE;
```

```
J 2
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
EDT$LWRITE
                                                                                                                                VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 (4)
                       EDT$LWRITE - write to a file EDT$$EXI_CMD - EXIT line-mode command
                                        RANGE [PREV_RANGE] = 0;
    788
789
790
791
792
793
794
796
797
                                     Position into the MAIN buffer.
                                         EDT$$FND_BUF (UPLIT (%STRING ('MAIN')), 4);
                                     Attempt to write the file, and set the EXITED flag if it succeeds.
                                         EDT$$G_EXITD = WRITE_FILE (RANGE, 1);
END;
                                                                                                        ! of routine EDT$$EXI_CMD
                                                                                       004D9
004DC P.AAB:
                                                                                                            .EXTRN
                                                                                                                       EDT$$FND_BUF, EDT$$G_EXITD EDT$$G_RCOV_MOD
                                                                                                                       EDT$$EXI_CMD, Save nothing #32, SP EDT$$G_RCOV_MOD, 1$ #2818, RANGE RANGE+20
                                                                                                           .ENTRY
                                                                                0000
CE8
B04
DD
FB
DDF
FB
DO4
                                                                                                                                                                                          1331
                                                                             2008FE04F201A200
                                                            00000000G
0802
14
                                                                                                                                                                                          1379
1384
1386
1390
                                                                                                            BLBS
                                                                                                            MOVW
                                                                                                            CLRL
                                                                                                            PUSHL
                                                                                                                       P.AAB
#2, EDT$$FND_BUF
#1
                                                                      E3
                                                                                                            PUSHAB
                                         0000000G
                                                                                                            CALLS
                                                                                                            PUSHL
                                                                                                                                                                                          1394
                                                                                                                       RANGE
#2. WRITE FILE
RO, EDT$$G_EXITD
                                                                      04
                                                                                                            PUSHAB
                                        00000000G CF
                                                                                                            CALLS
                                                                                                           MOVL
                                                                                                                                                                                          1395
; Routine Size: 50 bytes.
                                           Routine Base: _EDT$CODE + 04E0
```

: 798

1396 1

00

ED.

```
K 2
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
EDT$LWRITE
                        EDT$LWRITE - write to a file
EDT$$WR_CMD - WRITE line-mode command
                                                                                                                                  VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER: [EDT. SRC] LWRITE. BLI; 1
     800
801
802
803
                                    *SBTTL 'EDT$$WR_CMD - WRITE line-mode command'
GLOBAL ROUTINE EDT$$WR_CMD
: NOVALUE =
                                                                                                           ! WRITE Line-mode command
    804
805
806
807
808
809
810
                                      FUNCTIONAL DESCRIPTION:
                                               Command processing routine for the write command. Look at the current range; if it is null, use the whole buffer. Then write the file.
    812345678901234567890123456789
                                      FORMAL PARAMETERS:
                                               NONE
                                       IMPLICIT INPUTS:
                                               EDT$$Z_RNG_ORIGPOS
EDT$$A_EXE_CURCMD
                                       IMPLICIT OUTPUTS:
                                               EDT$$A_CUR_BUF
                                      ROUTINE VALUE:
                                               NONE
                                       SIDE EFFECTS:
                                               NONE
                                         BEGIN
                                         EXTERNAL ROUTINE
                                               EDT$$RD_CURLN;
                                         EXTERNAL EDT$$A_CUR_BUF : REF TBCB_BLOCK, EDT$$Z_RNG_ORIGPOS : POS_BLOCK, EDT$$A_EXE_CURCMD : REF NODE_BLOCK;
    8441234456789012345
844456789012345
                                                                                                          ! Pointer to the current command.
                                         LOCAL
                                               SAV_BUF,
RANGE : REF NODE_BLOCK;
                                                                                                           ! address of original buffer
                                         SAV_BUF = .EDT$$A_CUR_BUF;
RANGE = .EDT$$A_EXE_CURCMD [RANGE1];
                                                                                                           ! save original address
                                          IF (.RANGE [RAN_TYPE] EQL RAN_NULL) THEN RANGE [RAN_TYPE] = RAN_WHOLE;
                                          WRITE_FILE (.RANGE, 0);
                                      Reposition to the original line.
```

```
16-Sep-1984 01:03:36
14-Sep-1984 12:23:47
EDT$LWRITE
                                   EDISLWRITE - write to a file EDISSWR_CMD - WRITE line-mode command
                                                                                                                                                                                                  VAX-11 Bliss-32 V4.0-742
DISK$VMSMASTER:[EDT.SRC]LWRITE.BLI;1
                                                              EDT$$A_CUR_BUF = .SAV_BUF;
EDT$$CPY_MEM (POS_SIZE, EDT$$Z_RNG_ORIGPOS, EDT$$A_CUR_BUF);
EDT$$RD_CURLN ();
END; ! do the positioning
END;
      857
858
859
860
                                    1454
1455
1456
1457
                                                                                                                                                                    .EXTRN EDT$$RD_CURLN
                                                                                                                         007C 00000

9E 00002

00 00009

00 0000C

00 00013

95 00017

12 0001A

90 0001C
                                                                                                                                                                                     EDT$$WR_CMD, Save R2,R3,R4,R5,R6
EDT$$A_CUR_BUF, R6
EDT$$A_CUR_BUF, SAV_BUF
EDT$$A_EXE_CURCMD, R0
4(R0), RANGE
1(RANGE)
                                                                                                                                                                    .ENTRY
                                                                                                                                                                                                                                                                                           1399
                                                                                      56 00000000G
50 00000000G
50 04
                                                                                                                     0600004BE0226E0
                                                                                                                                                                    MOVAB
                                                                                                                                                                   MOVL
                                                                                                                                                                                                                                                                                           1445
                                                                                                                                                                    MOVL
                                                                                                                                                                    TSTB
                                                                                                                                                                                                                                                                                           1448
                                                                                                                                                                    BNEQ
                                                                                                                                                                                     #11, 1(RANGE)
-(SP)
                                                                           01
                                                                                      A0
                                                                                                                                                                    MOVB
                                                                                                                             90 0001C
D4 00020 1$:
DD 00022
FB 00024
DO 00029
DO 0002C
28 0002F
FB 00037
04 0003E
                                                                                                                                                                    CLRL
                                                                                                                                                                                                                                                                                           1450
                                                                                                                                                                                    RANGE
#2, WRITE_FILE
SAV_BUF, EDT$$A_CUR_BUF
EDT$$A_CUR_BUF, RO
#14, EDT$$Z_RNG_ORIGPOS, (RO)
#0, EDT$$RD_CUREN
                                                                                                                                                                   PUSHL
CALLS
MOVL
                                                                      FACD
                                                                                      CF
                                                                                      66
                                                                                                                                                                                                                                                                                           1454
1455
                                                                                                                                                                   MOVL
MOVC3
                                                       60 00000000G
                                                                                      00
                                                                                                                                                                   CALLS
                                                              000000006
                                                                                      00
                                                                                                                                                                                                                                                                                           1456
```

; Routine Size: 63 bytes, Routine Base: _EDT\$CODE + 0512

: 861 1458 1 : 862 1459 1 !<BLF/PAGE> EDT\$LWRITE EDT\$LWRITE - write to a file EDT\$\$WR_CMD - WRITE line-mode command 1460 1 END 1461 1 1462 0 ELUDOM 864 865 866

M 2 16-Sep-1984 01:03:36 14-Sep-1984 12:23:47

VAX-11 Bliss-32 V4.0-742 Page 25 DISK\$VMSMASTER:[EDT.SRC]LWRITE.BLI;1 (6)

! of module EDT\$LWRITE

PSECT SUMMARY

Name

Bytes

Attributes

_EDT\$CODE

1361 NOVEC, NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC, ALIGN(2)

Library Statistics

File	Total	Symbols Loaded	Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[EDT.SRC]EDT.L32;1 _\$255\$DUA28:[EDT.SRC]PSECTS.L32;1 _\$255\$DUA28:[SYSLIB]STARLET.L32;1 _\$255\$DUA28:[EDT.SRC]SUPPORTS.L32;1	377 9776 2	109 1 7	28 50 0 50	40 7 581 5	00:00.2 00:00.1 00:04.1 00:00.2

COMMAND QUALIFIERS

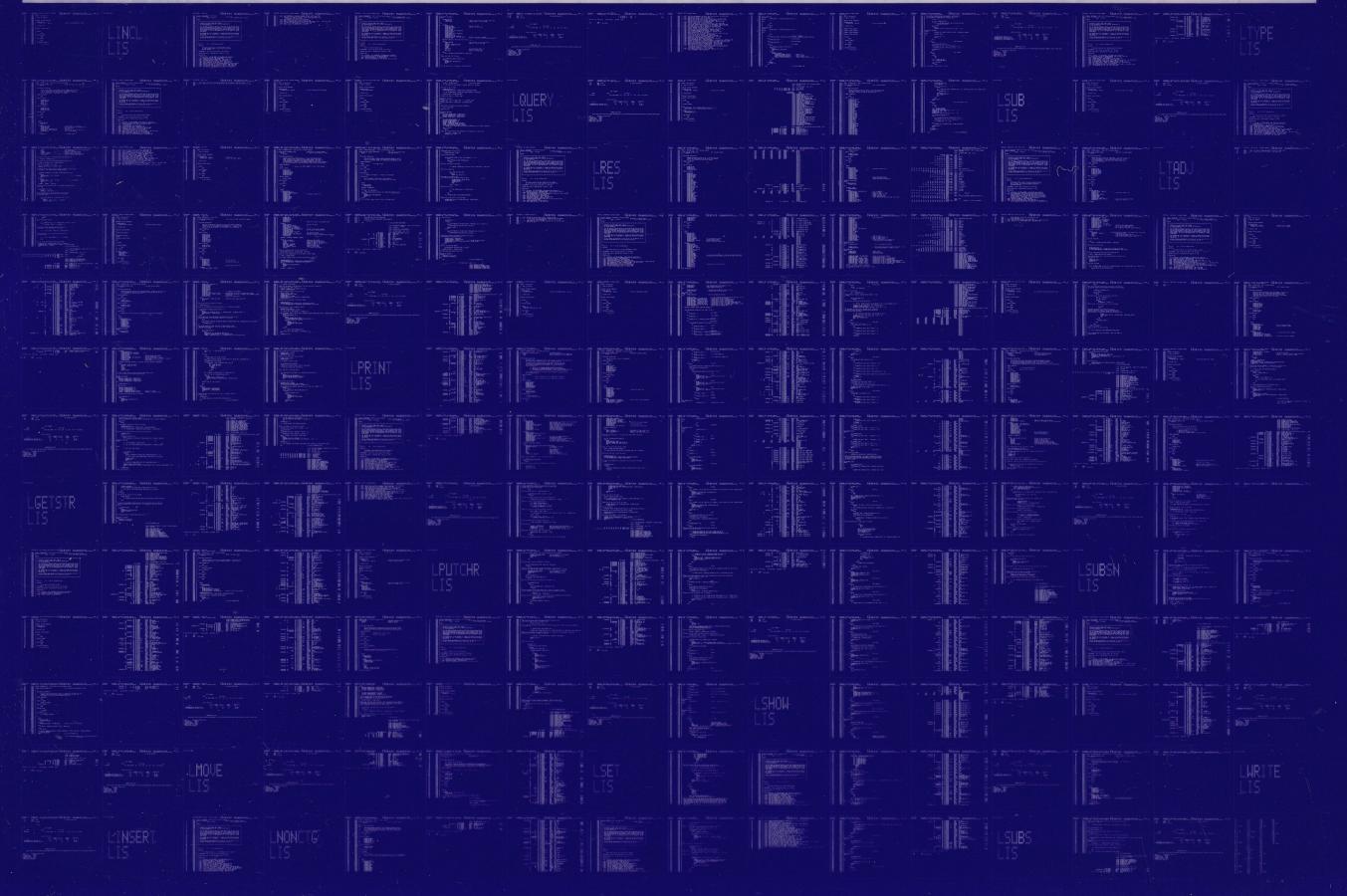
BLISS/CHECK=(FIELD, INITIAL, OPTIMIZE)/NOTRACEBACK/LIS=LIS\$:LWRITE/OBJ=OBJ\$:LWRITE MSRC\$:LWRITE.BLI/UPDATE=(ENH\$:LWRITE)

1346 code + 15 data bytes 01:00.3 01:15.8 : 1454 Size: Run Time:

; Elapsed Time: 01:15.8 : Lines/CPU Min: 1454 ; Lexemes/CPU-Min: 7386 ; Memory Used: 354 pages ; Compilation Complete

0136 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY



0137 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

